

JACG JACG

NEWSLETTER
Vol. 2 No.12

Aug. 1983
Single Copy
Price \$1.50

THE JERSEY ATARI COMPUTER GROUP

From the Editor's Desk ...

I am very pleased with this issue of the JACG Newsletter. In it you will find several firsts. The first of a multi-part article on programming languages from Rich Rospond. A new column entitled "Technical Corner" by Ernie Rice and another new column by Scott Brause on ATARI graphics. Also, I am continuing my regular column entitled "Tidbits". And, Curt Springstead is back with his PILOT column that we all have come to enjoy and benefit from.

Starting next month, I will begin writing a regular column for the Newsletter entitled "Art's Arcade". If you have enjoyed my game demos at the meetings then I think you will like what I have planned. All of the latest games will be covered in my usual "light-hearted" way plus a few surprises. Also coming next month (I think), is Dick Kushner's BASIC column. Everything you always wanted to know about BASIC but were afraid to ask, might be covered.

This is my 12th issue of the Newsletter and I am now seeing it become what I had envisioned originally. The Newsletter has come a long way (as have I) and I am continuing to strive for the best publication I can produce. I think having regular columnists that provide useful and timely information is important. Also important, is being able to count on these individuals to continue to support the Newsletter with their writing. YOU benefit from their expertise and THEY benefit from the experience.

These last few months have been extremely busy for me. I have been involved with several projects, some of which I cannot yet tell you about. In any case, I have enjoyed this last year as Newsletter Editor and look forward to the next year even more. There will be some changes coming to this Newsletter. Perhaps by this time next month I will be able to share some of the exciting news with you.

I would like to publicly thank Howard Johnson for his help in getting my ATARI up and running on UNIX. I am using Howard's "The UNIX Connection" that allows my ATARI computer to interact with UNIX at Bell Labs. This program was demonstrated at the last meeting and advertised in a recent issue of the Newsletter.

That's it for this month. Enjoy this issue. If you have any comments or suggestions, as usual, let me know. I am always willing to listen. Until Volume 3 No. 1, next month, I am—

Arthur Leyenberger
Editor in Chief, JACG Newsletter ▲

IN THIS ISSUE

Reviews

Atari Microsoft BASIC II
A. Leyenberger
Starcross by Infocomm
Eric Brown

Special Features

Comparing ATARI and LJK DOS
H. Johnson
Menu Program
J.M. Apice
Letter Perfect and Prowriter Revisited
J.F. Diglio
Coin Program
L & S Van Slyke

Regular Features

At a Turtle's Pace with Atari Pilot
Curt Springstead
Tidbits
A. Leyenberger
Technical Corner
E. Rice
Atari Languages
R. Rospond
Graphics Tablet
S. Brause

AND NOW A WORD FROM THE PRESIDENT....

Atari, Inc. has taken a step toward its new "lean and mean" image. It has reduced its white collar work force in Silicon Valley by 25% (from 4000 to 3000). Included in that cut is one-half of the User Group Support Team. We are now left with Earl Rice and Mark Cator who are now responsible for EVERYTHING including answering the telephones. Does this mean that Atari really doesn't yet understand the value of User Groups? That is certainly one possible interpretation of the data. Another is that User Groups support themselves just fine, thank you, and don't need corporate help at all. This would be a short sighted view on Atari's part. Looking at this with a little perspective is difficult. We'll have to see how things look six months from now. Maybe Atari will have Alan Alda visit all the user groups!

With all the problems I hear about the Commodore 64 computer in terms of production quality, we should thank Atari for keeping their construction standards high on the 400 and 800 models and hope that they keep it up with the new XL series of computers. I think that most of the people Atari would be interested in getting as purchasers will pay a little more for a quality product. Without large numbers, we User Groups are good testers of all new products that come out and can report quickly to our members and to other groups as to our experiences. Most of those who purchase the "throw-away" computers (under \$100), will eventually want to get a better machine or else stick the computer in the closet next to the game machine. So, it's really only those who get involved in computing who are around to make subsequent hardware and software purchases. These people are the "market" where money can be made.

Nominations for JACG officers for our next year (beginning in September) will be taken at the August meeting and during the rest of that month, with elections to be held in September. All the current officers are seeking re-election with the exception of Dennis Kushler, our Vice-President, who is moving to Michigan. Nominations will be accepted for all offices in August. If you're interested in a particular office, contact me.

NOTICE: Look for a terrific new book on Atari BASIC coming out near the end of this year. I am writing this book now and I'm very excited by what it does. It goes from the very beginning for new owners to some pretty advanced, but understandable stuff. It aims to teach programming on the Atari. I think it bridges the gap between general discussions of BASIC and detailed enumerations of the Atari commands and features. There are now several books out there on Atari BASIC and I am trying to avoid some of their mistakes and make the book understandable to all BASIC programmers. This is no mean

feat in a book that must go from "HELLO, I'M YOUR ATARI COMPUTER" to looking at display list modification. I'm enjoying working on this project and I hope that you'll give it a look when it appears. Needless to say, I'll have more to say on this subject in the months to come. I am also planning on using my experience in writing this book to put together a monthly column on Atari BASIC for the JACG newsletter, with the first installment hopefully coming in September.

Dick Kushner
JACG President.

JACG MEMBERSHIP

The Jersey Atari Computer Group (JACG) invites you to become a member. Dues are \$15.00 per year and entitle the member to 1) Receive the monthly newsletter and when you join, receive back issues of the newsletter as available; 2) Purchase programs from the group's extensive tape and disk librarys at special rates; 3) Join special interest groups or form new ones; 4) Benefit from the expertise and experience of other Atari computer users; 5) Participate in group purchases of software at substantially reduced prices; 6) Receive a membership card that entitles the member to discounts at local computer stores; 7) Attend monthly meetings to learn about the latest hardware and software, rumors, and techniques for getting the most out of your Atari computer; 8) Submit articles and programs to the newsletter and give demos and presentations at the monthly meetings; 9) Participate in sale/swap activities with other members; and 10) Have a lot of fun.

If all of this sounds good then send a check or money order, payable to JACG, to:

Rick Olson
5 Starling Drive
Randolph, NJ 07869

Remember, receiving the JACG Newsletter is just one of the many benefits of being a member of JACG.



GEMINI



YOUR ATARI CONNECTION

ATARI USER GROUPS ONLY — SALE ENDS AUGUST 30, 1983

EPSON

PRINTERS

RX-80

399.95

FX-80

549.95

MX-100

639.95

MX80FT

449.95

C. ITOH

8510AP

369.95

PROWRITER
PRINTER

SOFTWARE

VISICALC **159.95**

ATARI WRITER

79.95

MICRO SOFT

BASIC II

69.95

ATARI **800** 48K

\$269.95

DISK **810** DRIVE

\$419.95

1010 RECORDER

\$74.95

1025 PRINTER

\$399.95

COLOR I

AMDEK 13" MONITOR

\$284.95

HAYES

300 SMART MODEM

\$209.95

J-CAT

MODEM

\$119.95

MARK II MODEM

SIGNALMAN

\$74.95

DISKS

GEMINI

SS, SD

10/ **16.95**

ELEPHANT

No. 1 SS, SD

10/ **19.95**

MAXELL

SS, DD

10/ **24.95**

FLIP n' FILE

Holds

50 Disks

19.95

BASIC A +

59.95

TINY C

79.95

MAC/65

64.95

C/65

64.95

LETTER PERFECT

40/80 **119.95**

DATA PERFECT

40/80 **99.95**

GEMINI

GEMINI

GEMINI

86 RIDGEDALE AVENUE, CEDAR KNOLLS, NEW JERSEY 07927
STORE HOURS: Mon - Thur 9-6 P.M. Fri. 9-8 P.M. Sat. 9-4 P.M.
Closed Sunday

267-0988

TIDBITS
News and Views by
Arthur Leyenberger

Copyright (c) 1983 by Arthur Leyenberger

This column will bring you news, rumors, helpful hints and whatnot every month. If you have any information that you would like to share, send it to me and I will use it.

ATARI 1450XLD Delay

I recently heard, from a very reliable source, that the new ATARI 1450XLD computer recently announced at the June CES in Chicago (and mis-labeled as the 1250XLD on the front page of the July JACG Newsletter) will not be ready by Christmas. This computer is the top of Atari's new line of computers and contains a built in voice synthesizer, direct-connect modem, 64K RAM and a double-sided, dual-density (127K bytes) drive that is connected directly to the processor bus for a mind-boggling access speed of 2-3 times that of the venerable 810 drive. We of the Atari ilk were hoping that the entire line (600XL, 800XL, 1400XL and 1450XL) would make it out by Christmas as promised. Although there is always a chance that Atari will meet the Jan. 1 deadline, the fact that this rumor is occurring now, 5 months away from the promised date, is not good news. I will keep you posted in this column. Now lets go to the video tape!

RANA, RANA, Where are you?

Conspicuous by its absence, the RANA 1000 magic disk drive has been in transit (en route to dealers) for over a month now if we are to believe RANA corp. in Texas. RANA has promised many people that that drives "were shipped" in mid-June. However, the closest anyone has ever gotten to an actual piece of hardware was depicted in the centerfold of the July JACG Newsletter - the RANA box. I was able to obtain a RANA box for evaluation and you will be reading my review of it in the September issue of this yellow journalistic rag. In the meantime, I can only say that the BOX performed flawlessly in all of the tests that it was subjected to. The BOX is a stand-alone device which has a user-friendly human interface. On-line instructions are printed right on the BOX itself, as are useful help messages such as: "This END UP", and "Do not stack more than 5 high". The BOX is really a multi-purpose device (those clever RANA engineers) since, in addition to housing the hardware during non-shipment and when the merchandise is not on retailers shelves, the BOX functions very nicely as a wastepaper basket, kitty-litter recepticle and storage compartment. If the actual hardware (if it ever is produced) is as good as the BOX, RANA will do quite well.



Editor's Desk Exposed!

When I am not writing late-breaking news stories (see above) I am also the Editor of this Newsletter. Many people have asked me about the column that I put on the front page of every issue called "From the Editor's Desk". It seems that these folks, and maybe even you, believe that there is no such thing as the Editor's Desk. In fact, they wonder if there is really even a desk at all, and who does it belong to, for goodness sake. Well, to put this issue to rest once and for all and avoid any further controversy, you will see either below or to the right an actual, rare picture of the Editor's Desk. As you can see it isn't really a desk, but an interactive, on-line, multi-task Editor's Work Station. However, you would think I was pretty stupid if on the front page of every Newsletter it read, "From the Interactive, On-line, Multi-Task Editor's Work Station". 'Nuff said.

That's it for this month. Actually, not too much has been happening in the world of Atari. Next month, I will get real serious and tell you about Planet Missionary II. Just when you thought it was safe to believe everything you read in the papers!



```
*****
*****
***                                     ***
***  #####  ##  #####  #####  ***
***  #####  #####  #####  #####  ***
***  ##  ##  ##  ##  ##  ##  ***
***  ##  #####  ##  ##  ##  ***
***  ##  ##  #####  ##  ##  ##  ***
***  #####  ##  ##  #####  #####  ***
***  #####  ##  ##  #####  #####  ***
***                                     ***
***                                     ***
***  BULLETIN BOARD  ***
***                                     ***
***  24 HOURS/DAY  ***
***                                     ***
***  201-549-7591  ***
***                                     ***
*****
*****
```


Comparison of Atari DOS and LJK DOS
By Howard C. Johnson - JACG

Recently I wanted to be able to convert files from LJK Letter Perfect (r) to and from ATARI DOS. Rather than buy the conversion program I decided to write my own. This would provide an excuse to find out how LJK DOS worked and a first application of the Deep Blue C compiler (APX-20166).

ATARI DOS is designed for easy sequential access to data in files. (E.g., loading a program.) It doesn't handle random access at all well. On the other hand, LJK DOS is optimized for random access.

DISK FORMATS

The ATARI DOS is well documented in "Inside ATARI DOS", by Wilkinson. Briefly, each file is composed of sectors linked together. This is achieved as follows:

- a. The directory entry gives the address of the first data sector.
- b. Each data sector identifies how many data bytes are in it (usually 125), and the location of the next data sector.

This is what makes random access so difficult. Every sector has to be read in order to locate a given byte number.

LJK DOS only puts data in a data sector. Each sector contains 128 bytes except for the last sector in the file. In order to locate the data sectors, extra allocation sectors are associated with the file. Each of these sectors contains a list of the 60 data sectors it controls. The allocation sectors are linked, and the last one may be short. Seeking a particular byte in a file will be 60 times faster using LJK DOS.

HOW LJK DOS IS BUILT

The LJK DOS bit map starts at sector 362 and the Directory uses sectors 363 through 371. The structure of the bit map and directory are identical to ATARI DOS. LJK DOS utilizes lower case file names and upper or lower case extensions. LJK Letter Perfect formats the disk to make LJK DOS ignore sectors 8-55. These apparently are used by the printer driver.

The allocation block contains an 8 byte header, then a list of 60 data files. The Header is comprised as:

Byte 0: \$02.

Bytes 1,2: Sector number of the next allocation block. \$0000 if this is the last allocation block.

Byte 3: \$01.

Byte 4: \$10 in the first block.
In each successor block, add \$1E.

Bytes 5-7: Three byte, byte count of the file. This count is duplicated in each allocation block.

The balance of the record (bytes 8-127) contain up to 60 sector numbers, stored as 2 byte quantities.

TWO DOS's ON ONE DISK

My conversion program takes advantage of the fact that both ATARI and LJK DOS may be on the same disk. This results from the fact that ATARI DOS uses sectors 360 and 361, while LJK DOS will use 362-371. This permits up to 8 ATARI files to be on the disk. Over 8, ATARI overwrites sector 362, the LJK bit map!

USING C

A program like this is really writing a Disk Operating System. By using a structured high level program such as C, this can be done in pieces and then integrated. Debugging is simplified because each module has a unique function and C permits adding PRINT statements as needed throughout the modules.

Finally, this activity is like solving a mystery and changes have to be made as one develops the program. C permits individual modules to change with minimal impact on the balance of the program. △

Shamus:Case II

Reviewed by:Jon Harrod - JACG

It's here! Yes, fellow Shamus fans, Case II is now on the market. Thrill to the continuing adventures of our large-nosed detective friend. But this time it's going to be a little harder to kill the Shadow (he's back again).

Yep, the Shamus is still up to his old tricks, but now he's under water and he has no knives! In Case II you must get to the Shadow by jumping snakes, killing whole rooms full of mutant crustaceans, grabbing keys and opening doors, and avoiding spiked pits.

Some of the nice things about the game are the colors (a step up from that sickly yellow), having a map that builds itself as you go from room to room, and an ally, in the form of a bird. When you shoot it a certain amount of times it turns to a fireball to help you destroy some of the strange creatures. Some of the more sadistic features of the game are slider-rung ladders, on which the rungs may disappear, the fact that the mutant creatures can make you fall down a level, and tempting passages that you can't fit in.

If your like me, and like Shamus (who doesn't?), then your almost guaranteed to like Shamus II. So go take a look at it and all the other new Synapse games, and take your pick!

Letter Perfect and Prowriter Revisited

By John F. Diglio - JACG

First I would like to thank Art Leyenberger for leading me to LJK's Letter Perfect and getting me started with my Prowriter (AKA-NEC 8023, C.Itoh 8510, Sony SMI-7020 etc.). This article is intended as a follow-up to his tutorial from the February '83 printer issue of the JACG newsletter.

Lets try a quick review of how to start up the system: 1. Turn on the disk drive. 2. Put in the disk marked "LETTER PERFECT ATARI". 3. Boot this to the computer. 4. At the (EADQ) prompt hit return (for now), File Unit prompt hit return, Data Base Unit prompt hit return. 5. Remove LETTER PERFECT disk from drive and select "format disk" from main menu. 6. Put in a blank disk and follow prompts. 7. Remove formatted disk and insert "LP ATARI Disk printer editor" and boot (open and close hatch or turn computer off and on). 8. At menu enter 6-return, page formats - change to suit your needs. HIT AN EXTRA RETURN to get next menu. 9. Boldface double width(y/n): hit return (do not enter a value). 10. Backspace character: 0 return (do not enter 8 - enter 0). 11. At next menu enter these responses: 1.)...: 27,33 2.)...: 27,34 3.)...: 27,88 4.)...27,89 at 5&6 no entry should be made. HIT AN EXTRA RETURN to get next menu. 12. enter 4 fonts, I usually use 1. Font 1: 27,78 2. Font 1:27,69 3. Font 2:27,81 3:Font 3:27,80. These are pica pitch (10 cpi), elite pitch (12 cpi), compressed pitch (17 cpi) and proportional. HIT AN EXTRA RETURN. 13. Now insert the LETTER PERFECT formatted disk and press - shift #. This is your data disk. 14. Boot "LETTER PERFECT

ATARI" again and at (EADQ): hit D - insert formatted data disk and press - shift #. File unit - return, Database unit - return. 15. Select editor from main menu and begin creating.

To answer some of the questions left open in the original article lets look at a feature of Letter Perfect, the ability to pass any control code to the printer. This function is initiated by holding the control key down and typing the letter U (I will use an underlined U as this character. Try elongated letters, like in the title of this article. As the manual of Letter Perfect suggests us format lines (control F followed by the codes you want to change, page 33 of the manual). In this line choose the font you want to type in. If you want some of the text elongated (double width) just enter U(14), to return to normal type enter U(15).

This printer can not print super or subscript letters but it can print superscript numbers, as you would use in footnoting. I called Leading Edge and asked if this could be done and they said no. Lets try the impossible, first put the toggle on for Greek Symbols by using U(27)U(38). This can be done anywhere in the text before you need to use the superscript number for a footnote or power or even degree sign. An example is:

I have the quote "58
or 98.60F.
or 2³=8.

How do we do these the 58 was U(212)U(215) the 0 was U(207) the 3 was U(210). The control codes are 0=207, 1=208, 2=209, 3=210 etc. to 9=216. The same method is used to get any of the Greek letters or the % and %. Now the charts in the users manuals can be used to get any symbol code passed to the printer while using Letter Perfect.

Continued on

Technical Corner

By Ernie Rice - JACG

Welcome to the technical corner of the JACG newsletter. My name is Ernie Rice, and I am the President of EHR3, Inc., a corporation devoted to software development for the ATARI personal computer systems. I have been asked to write a column for the newsletter every month. Topics to be presented will be of a technical nature, dealing with subjects such as ATARI's DOS, BASIC, OS and other technical areas. I will also touch on advanced programming techniques in both Assembler and BASIC.

I have been given a 'free hand' in this column, as a result I encourage input from the user community as to which topics they would like to see presented. I hope to make this column informative for all levels of programmers, not just the 'heavies'.

Anyone who would like to submit ideas for this column are invited to do so by either writing to me at :

Ernest H. Rice III
EHR3 Inc.
174 Summit Ave
Summit, NJ 07901

or by calling (201) 277-6785

Also feel free to call me for any question about the ATARI. I don't assume to know all the answers, but I promise to do my best at researching all inquiries.

In this month's column, I will describe the infamous 'No list' protection scheme available to those who program in BASIC. This scheme is quite effective in protecting a user of proprietary BASIC programs from displaying the source code. Until recently, there was no way to remove the protection once it was added to a module. A user could issue a RUN command for the desired module and all would be fine. If, however, the user attempted to LOAD the file, the next immediate command entered (like LIST) would cause the system to hang-up.

The process of adding this protection scheme is as follows:

1. The author writes a program in BASIC
2. Once the author is sure the module is fully tested, he/she makes a backup of the file. This is imperative as there has been no way to remove the protection scheme--- until recently.
3. The author adds one line of code (listed below) to the END of the program to be protected.
4. By issuing a GOTO 32767, immediate command, the user causes the protection

scheme to be implemented, along with a saving of the protected version to disk.

5. The user now has a produced a protected version of the BASIC program. A user may say RUN D:PROGRAM but a LOAD "D:Program" followed by any command will cause the system to hang up.

As I said earlier, there has been no method of removing this protection scheme, until recently.

John Anderson, of Outpost ATARI fame, and I have had several discussion about how this protection scheme worked. Also whether or not it was really permanent. A bet ensued, and I felt that since a modification was made only to the file on disk which was loaded, that you should be able to modify that file directly to remove the protection.

I sat down at my 800 one Sunday afternoon and wrote the program listed below. It examines a BASIC program on disk which was protected via the above method, and modifies the program so that it can be listed once again.

The process is pretty straight forward. The program looks for a specific BYTE in the module, calculates a value from the remaining data and replaces the original value. The BASIC program to remove the protection scheme is not very graceful. It could be reduced to only a few lines of code; however, it does the job. The bet was won.

Why am I revealing the secret of this protection scheme? Simple- I feel that this is not a really good protection scheme. There are much better ways to protect your code.

For example, destroying the variable table will make it very difficult for a user to examine any complicated program. This protection scheme, when incorporated with bad-sectoring and other methods, seems like a more viable solution. The user can not just change one BYTE in the module to return to the original source code. Bad variable names may be reconstructed, but not with the descriptive names they originally had. Trying to follow someones logic in this case would be difficult at best. Another reason for revealing this scheme is that many users who tried it forgot to backup their original code. As a result they no longer have access to their own programs (other than running them). The publication of the 'secret' will return the code to its rightful owner and force the manufactures of software to use better portection techniques.

This scheme can also be found in the September issue of Creative Computing magazine in the 'Outpost ATARI' section.

ALISTING #1:SOURCE TO PROTECT A PROGRAM
LOAD THIS PROGRAM AND ADD THIS LINE.
THEN SAY 'GOTO 32767'

AT A TURLTE'S PACE with ATARI PILOT
by Curt Springstead JACG
Copyright Curtis Springstead 1983

One of the hottest uses of personal and home computers is adventure gaming. Educationally these games can be used to teach and develop deductive logic, spatial relations and visualization (trying to picture the layout in your mind from the descriptions given) and reading skills. Creating these games can help with the mechanics of grammar and creative writing.

What I would like to do in this column is start a discussion of the mechanics of an adventure game program and specifically the heart of these games the parser or interpreter as I prefer to call it. This is the part of the game which interprets the instructions you type in and tells the computer what to do about it.

For those of you not familiar with adventure games let me briefly tell you what they are about. The author has created an environ, from dungeons to outerspace, and asks you to enter this "world" to execute a specific task or tasks. To do this you tell the computer through typed instructions what you want to do and where you want to go. The machine, if it understands, then presents in narrative form the result of your action, which could be a description of a new surroundings, the result of a combat situation or the result of a puzzle solving attempt. What separates the men from the boys in these games is the extent of the vocabulary and the ability of the interpreter program to translate what you type into commands the program was designed to understand.

Most of these games rely on a table look up method of interpreting the commands, for our purposes here that will do for us also. This means that the words entered are matched to the list of words in the vocabulary of the program or game. If the word is found it is then classified into noun or verb class, is it a thing or an action. For each verb in the game there is a portion of the program that carries out that action. For example, if you type GO NORTH the system checks to see that it understands GO and NORTH and then tries to do a GO action. In the GO section of the code there will be a break to a section of the GO routine that defines how or where the system GOes NORTH. This may point us to a new room or place and then cause the system to know that we have moved and that it must display a new description of the surroundings.

Included here is a starting point for your adventure program. This program demonstrates a simple method of interpreting the commands and making the actions indicated. By adding more options on the match list for the room, such as instructions for GO UP and GO DOWN you can create movement in several levels. Setting up matches for the other "verbs" will allow the adventurer to do more than move around. You can devise an inventory system which will allow the rooms and the player to have items in their possession.

```
10 C:$BLANK=
20 C:$ERROR=
30 J:$ROOM1
40 R: ASK PLAYER FOR COMMAND
50 *COMMUNICATE
60 T:WHERE TO NOW?
70 A:$BALANCE
80 *PARSE
90 A:=$BALANCE
100 MS: , Enter ESC CTRL Cursor Right Space
110 J(%M=0):*ACTION
120 C:$WORD=$LEFT
130 C:$BALANCE=$RIGHT
140 A:=$WORD
150 M:GO,LOOK,GET,DROP
160 CY:$VERB=$WORD
170 JY:$*PARSE
180 M:UP,DOWN,EAST,WEST,NORTH,SOUTH
190 CY:$NOUN=$WORD
200 J:$*PARSE
210 *ACTION
220 C:$COMMAND=$VERB $NOUN
230 A:=$COMMAND
240 E:
```

```
250 R: * * * * * ROOM 1 * * * * *
260 *ROOM1
270 T:) Enter ESC SHIFT CLEAR
280 T:
290 T:$ERROR
300 T:YOU ARE IN ROOM 1
310 T:THERE ARE TWO DOORS
320 T:ONE ON THE EAST WALL AND
330 T:ANOTHER ON THE SOUTH WALL.
340 U:*COMMUNICATE
350 C:$ERROR=
360 M:GO NORTH,GO SOUTH,GO EAST,GO WEST
370 JM:$NOG01,*ROOM3,*ROOM2,*NOG01
380 C:$ERROR=SORRY I DONT'T UNDERSTAND
390 J:$ROOM1
400 E:
410 *NOG01
420 C:$ERROR=CAN'T GO THAT WAY
430 J:$ROOM1
440 R: * * * * * ROOM 2 * * * * *
450 *ROOM2
460 T:) Enter ESC SHIFT CLEAR
470 T:
480 T:$ERROR
490 T:YOU ARE IN ROOM 2
500 T:THERE ARE TWO DOORS
510 T:ONE ON THE WEST WALL AND
520 T:ANOTHER ON THE SOUTH WALL.
530 U:*COMMUNICATE
540 C:$ERROR=
550 M:GO NORTH,GO SOUTH,GO EAST,GO WEST
560 JM:$NOG02,*ROOM4,*NOG02,*ROOM1
570 C:$ERROR=SORRY I DONT'T UNDERSTAND
580 J:$ROOM2
590 E:
600 *NOG02
610 C:$ERROR=CAN'T GO THAT WAY
620 J:$ROOM2
```

Continued on 17



EARTH STATION

Although the ATARI computer is a personal computer, and is best known for its graphics and sound capabilities, it offers one of the best choices of computer languages available for the home user. In addition to several implementations of BASIC, ATARI also offers Pascal, C, PILOT, LOGO, FORTH and Assembler. This offers an opportunity for any ATARI owner to get a solid foundation in programming languages.

Over the next several months, I will use BASIC as the basis for analyzing the other languages that are currently available. In order to do this I will ignore specific implementations of a given language, and concentrate on the essentials that separate the languages from each other. So if you are wondering what the difference is between BASIC and Pascal or LOGO, then I think these articles are for you.

What is a language?

Before 1954, almost all programming was done in machine language. At that time the work of writing instructions was called coding rather than programming-programming implies the more difficult task of designing algorithms. Machine language is very fast in execution, but extremely time consuming for those writing the program.

Then in 1954 John Backus formed a group to develop the FORTRAN compiler which would translate mathematical formulas into machine language. The aim of the group was to greatly shorten the amount of time necessary to prepare a program. They did not address such subjects as machine interdependence, block and control structures, etc.

In the late 1950's, other languages began to appear, and in 1959 the designers of COBOL addressed the issue of a machine independent language that could be used by the business community. It would be hard to describe the complexity of their task, and in developing the earliest 'higher level' languages there was an absence of 1. economy of design and 2. a natural means of expression.

These concepts are taken for granted by users of today's more sophisticated languages, but are extremely important for those of us who have written in those original languages.

There are several categories of programming languages:

A commercial language is concerned with the production of reports. COBOL is the best known commercial language.

A scientific language is used mainly for

the manipulation of numeric data. FORTRAN is the best known scientific language.

An interactive language allows the programmer to make changes and corrections from a terminal during execution. Lisp and APL are examples of this.

A procedural language allows the user to specify a set of imperative statements that are to be performed in a particular sequence. Most contemporary languages are procedural.

There are other categories that are less important for our purposes.

Implementation Schemes:

Programming languages may be implemented in one of two ways- compilation or interpretation. This is a concept that we see with the ATARI.

In compilation the program is written in the programming language, the SOURCE program, then translated into an OBJECT program, which is the machine language of the computer. Both Pascal and Microsoft BASIC are compiled languages.

With interpretation the source program is translated into an object program that can not be executed directly by the actual computer. Instead, the execution of the object program is achieved by an interpreter. This is a program that is executed on the target machine performing the operations specified in the object program by means of subroutines.

In simpler terms- in Microsoft BASIC we write and SAVE the program, which is then compiled. Any errors are listed after we run the program. In ATARI BASIC errors are listed immediately after we complete each line. An interpreted language gives us immediate feedback as to syntax errors, but a compiled language executes faster. In business applications, almost everything is compiled due to the demand for processor time.

An analysis of BASIC:

BASIC is the most widely used language for personal computer systems. It was developed at Dartmouth College as a teaching language using elements of both FORTRAN and ALGOL. There are many versions of BASIC. For ATARI we have Microsoft BASIC and ATARI BASIC with all the add-ons such as Monkey Wrench, etc.

BASIC uses line numbers during execution of the program. If a line number follows RUN, then execution starts with that line number, if not, then execution begins with the lowest numbered line. The program follows a numerical sequence until a STOP or END statement is encountered.

Variables are introduced as needed, and there is no need to declare them before

Hardware

Software

MICRO CONCEPTS

"Atari® for Less"



NICK SCALERA
(201) 328-0752

AUGUST JACG

SPECIALS

MOASIC 32K RAM BOARDS \$75.00

CENTIPEDE \$30.00

ELEPHANT DISKS \$20.00

COMPUSERVE STARTER KIT \$30.00

SALUTE TO ATARI By Scott Brause - JACG

Before I begin, I would like to say that I am not an employee for ATARI. I am going to do something very strange (alot of people will probably think I am crazy). I am actually going to praise ATARI. Most people feel that ATARI has done alot wrong. I cannot argue with that but, they have also done alot right. Especially in the area of customer relations. I recieved my 800 one year ago (after selling a 400) and since then I have received nothing but help from ATARI.

When I first started out, I knew nothing about how to program. I got help from a friend (Bob Applegate). But, sometimes I started doing things that we couldn't figure out how to do. So I began calling ATARI's toll free number (800-538-8553) to ask for assistance. Their customer relations people were always very nice and took my name down and the problem I was having. Within 2 days I usually get a call back with someone trying to help. Even though they don't always know the answers they get an A for effort. And just about each time I call I get a set of DEMOPACS. These are put together by ATARI programmers showing examples of how to do various things. Even though they cannot help me with my problems any more it shows an advantage of owning an ATARI.

With the introduction of the new ATARI models we shall see a much larger percentage of ATARI users and alot better support. ATARI, I salute you!!!!

JERSEY ATARI COMPUTER GROUP (JACG) MEMBERSHIP FORM

PLEASE FILL OUT FORM COMPLETELY AND INCLUDE MEMBERSHIP FEE (\$20)
MAKE CHECKS PAYABLE TO JERSEY ATARI COMPUTER GROUP

SEND TO: RICK OLSON
TREASURER, JACG
5 STARLING DRIVE
RANDOLPH, NJ 07869

___ NEW

___ RENEWAL

NAME _____

ADDRESS _____

CITY/STATE/ZIP _____

PHONE NUMBER _____ HOME _____ WORK _____

EQUIPMENT: ATARI 800 _____ ATARI 400 _____ MEMORY _____ K

410 RECORDER _____ DISK DRIVE _____ MODEM _____ PRINTER _____

COMMENTS: _____

DUES ARE FOR SEPTEMBER 1983 TO AUGUST 1984
SEE NEWSLETTER ARTICLES FOR MORE INFORMATION

by

At a recent club meeting, I purchased several disks from the user library containing a variety of programs and utilities.

Many if not all of the library disks are self booting, using an AUTORUN.SYS file which calls a program named MENU.SYS written by E. R. Lieberman and Matt Loveless. The menu program previews the disk directory and allows the user to execute any resident file in either of two ways. The user may select the number preceding the filename and press the RETURN key, or he/she may use a joystick to position a cursor next to the file name and then press the trigger (fire) button to execute it.

My first thought was to install the MENU.SYS program on all my disks, since this seemed to be a good way of previewing my directory without going to DOS.

The enthusiasm quickly became a disappointment when I tried to load one of my binary files. Binary files are those machine language files that are normally loaded from DOS using the 'L' option. In examining the listing, I noted that the menu program had no provisions for this. I also noted that the joystick routine was not really necessary. In fact it took longer to use the joystick to position the cursor than it did to select a number and press RETURN. (Boy, you just can't please some people!).

I do not want to sound like I'm criticizing the program. I'm not. In fact it is well written and very useful. I did however, make some changes both aesthetic and functional.

The first change was the addition of a subroutine taken from an article which appears in COMPUTE!'S SECOND BOOK OF ATARI titled "Loading Binary Files from Basic" by Robert E. Allegher(with permission of course). The routine allows the user to load and run binary files from the menu with the only limitation that they will not occupy the free ram area in page 6 (1536 to 1791 decimal).

In order to keep the program short the joystick routine and header data were

removed (sorry Matt)

The program was then given some aesthetic changes to the default colors, from the traditional dark red and yellow to a subtle blue. The resulting program is listed below

It should be noted that, if the disk on which MENU.SYS resides contains more than 32 files, the menu will display the first 32. In order to view the remaining files you simply enter an "N" for next group, otherwise type the number adjacent to the program you wish to run.

Continued on 14

 THE GRAPHICS TABLET

BY
 SCOTT BRAUSE - JACG

Welcome to the first installment of THE GRAPHICS TABLET. This column will appear monthly, dealing with several aspects of ATARI graphics. The ATARI computers are light years ahead of most of the other computers at a fraction of the cost. This column will help those out that want to learn about these capabilities. Some of the topics I will be covering in upcoming columns are display list interrupts, animation, scrolling, page flipping, GTIA modes, mixed display lists, player missile graphics, XIO, and much more.

All programming in the next few months will be done in BASIC. But, if there is enough of a demand (I think and hope there is) then I will start including assembly routines.

The ATARI is an amazing machine and there are many things that can be done with it's capabilities if you have the information to do it. Unfortunately, there is no one book that can explain it all. That is why I am writing this column. If anyone has any questions please write to me at the address at the end of this column and I will try to answer them in the upcoming newsletters.

Some people have asked me questions about many of the commands found in the BASIC REFERENCE MANUAL. So, This month, I am going to try and give a better explanation of the ATARI graphics commands. For those of you that know these commands, be patient. I will get to more advanced things soon. I hope to parallel this column with a tutorial at the beginning of every meeting.

First the commands are listed, then an abbreviation(which can be used from basic), then a example of how the command is used and then the definition.

GRAPHICS (GR.), GR. N - Where N is a graphics mode from 0 to 15. This command puts you in one of the 11 (15 on the ATARI 1200XL) graphics modes on the ATARI computer. These modes are:

MODES 0-2 - Five color text modes. They can only be used to print text on the screen. Graphics mode 1 characters are larger then graphics 2 characters. Upper case can only be printed without modifying a memory location. By trying to print lower case and inverse you can get differnt colors.

MODES 12-13(1200XL only) - These modes are text- only modes. The character set must be modified to use them.

Modes 3-11 & 14-15 - These graphics modes are used for graphics only. Text can only be entered in the text window. As the number of the mode gets higher the resolution gets better.

Adding +16 to any of these modes will remove the text window and adding +32 will prevent the screen memory from being erased?

SETCOLOR (SE.), SETCOLOR X,Y,Z - This command lets you specify the colors to be used. The command is in the form. SETCOLOR X,Y,Z. X is a color register from 0 to 4. Y is a color value from 0 to 15. Z is a luminence from 1 to 14 (14 being brightest). Look on page 50 in you BASIC REFERENCE MANUAL for color values.

COLOR (C.), COLOR X - This command lets you choose the color to be used for all plot commands until the next COLOR statement is executed. In text modes the color can be a number from one to 255. After using COLOR the plot command and a letter would be displayed corresponding ATASCII number you specified. In a MAP MODE the color number corosponds to a color register. In different modes however the color register number may be different then the one you specified. For example in modes 3,5 and 7 if you specify COLOR 1 you will be referring to color register 0. Look on page 53 of THE BASIC REFERENCE MANUAL to see what COLOR commands to use in what mode.

PLOT (PL.), PLOT X,Y - This command will put one pixel group (pixels are each point that can be displayed on the screen) at the screen position X,Y. The color will be chosen by the last COLOR command.

DRAWTO (DR.), DRAWTO X,Y - The DRAWTO command will draw a line from the last plotted point to the coordinates specified by X,Y. The color is the same as the last point plotted.

POSITION (POS.), POSITION X,Y - This command is used to position the cursor at a certain point X,Y on the screen.

PRINT (?), PRINT X, PRINT "XXXXX", PRINT #6:"XXXXX", PRINT #6, X - This command is probably one of the most usefull commands in ATARI BASIC. PRINT X will print the value of a variable of string on a text screen. PRINT "XXXXX" will print the contents of the parenthesis. The other two will do the same thing except a little differently. These commands are used in conjunction with the POSITION command and they are used to print to a text screen (modes 1,2,12,13). By specifying #6 you are telling it to send the information directly to the IOCB (input output controll block) that is opened for the screen.

PUT AND GET (PU./GE.), PUT #n,X, GET #n,X - These commands are used to PUT and GET one byte data from the screen. First the cursor must be moved to the location on the screen where the data will be put or retrieved. Then execute the command. N is the number of the IOCB that data is to be put or retrieved from. Using PUT it will print the value of the variable somewhere on the screen (in a text mode if the screen is specified) or it will put a pixel on the screen (in a map mode). Get will retrieve information from the screen. The

Graphics Tablet

variable will hold the value of the data. The PRINT and INPUT commands can also be used to take in more data. If you use the PRINT command after executing a PUT or GET command, the pixel where the data was retrieved from will be changed. This also happens with the LOCATE command. To overcome this problem, before printing, use the PUT command and put the data that was read back in place.

LOCATE (LOC.), LOCATE X,Y,Z - This is one of the most misunderstood ATARI graphics commands. It is used like a GET without having to use the POSITION command. X and Y position the cursor at a certain place on the screen. The value of that pixel is stored as the variable Z.

XIO (X.), XIO 18,#6,0,0,"S:" - This command is a very hard command for some people to figure out. Instead of expaining it here next month I will give an example of what it is and how to use it.

I hope this guide will be a help to some people. I suggest you save it for later reference. If you have any questions or would like to see something covered in THE GRAPHICS TABLET talk to me at a meeting or write me a letter at:

12 BRADFORD ROAD
EDISON, NEW JERSEY 08820

Only your comments can help make this column better. Happy plotting!!!

EUREKA

by

Bob Nadler - JACG

Copyright Bob Nadler 1983

I hardly ever consciously discover anything. I'm at my best walking my dog, or half asleep. That's when things seem to dawn on me. My latest revelation came, however, while I was seated before my trusty 800, punching code for a new label for the ZX81-cassette version of my DARKSTAR™ program.

Before I get to my most recent blinding insight, though, let me tell you about stubbornness. For over a year I refused to upgrade my Epson MX-80F/T to Graftrax Plus, because my printer was still in warantee when the Graftrax Plus ROM was announced. I wrote to the Epson marketing organization expecting to be treated by them at least as well as the Atari people treated 800 owners whose machines were still in warantee when the GITA chip and the ROM Mod B chips came on the scene. No luck though. Epson's attitude was basically, tough! They told me if I wanted the new ROM chips I'd damn well have to buy them, at full retail, from a dealer. There was no way they'd supply them at cost.

So for a year I made do with a lack of sub-and superscripts, underlining, italics and the rest of the bag of tricks that

goes with Graftrax Plus. When, finally, thoughts of losing my trademark protection (being unable to put the superscript TM after my program's name on computer-printed labels might just do it) drove me to overlook my dislike for the Epson marketeers, and spring for the three ROMs. (Gemeni has a very reasonable price for the retrofit kit, though it's certainly not manufacturer's cost.)

Once the printer was modified (it took me about 15 minutes, working slowly) and tested, I ran amok trying out the seemingly endless combinations of type faces and printing tricks possible with the reworked machine. I found that I could easily insert escape and control sequences directly in the body of the text (after formatting) when using the Atari Word Processor. (I love the Atari Word Processor. I've written three books

with it and intend to use it to do many more. Why does everyone, including Atari, seem to hate it?) But I also found that I had to stand on my head to get the damned printer to do its stuff when working from Basic print statements. There is nothing less thrilling than typing -- CHR\$(27);CHR\$(45);CHR\$(1); -- just to tell the silly printer I want an underline begun. Then, to add insult to injury, I have to tell the printer -- CHR\$(27);CHR\$(45);CHR\$(0) -- to stop underlining.

I gritted my teeth and put up with this sad sort of arrangement for as long as it took me to finally notice that it is completely possible to insert both escape sequences and control characters directly in Basic print statements. All you have to do is keep them inside of the quotation marks.

Now when I want to underline in a Basic print statement, I simply set up the statement to give me exactly the line I want, but without the underline. Then I use control/insert to open up some space within the quotation marks, immediately before the word or words to be underlined, and tap the escape key twice. That gets me an escape character printed on the screen. I follow that by pressing the hyphon (-) key. Then I press control/A. That's only five key strokes. Next I move the cursor to the end of the word or words I want underlined, insert some space, and press escape, escape, hyphon, control/comma (,). That's all there is to it. The printer control characters don't mess up the line spacing or placement. The same system, of course, also works for any of the other printer commands (bold, italics, expanded, etc).

If you've been endlessly typing CHR\$ this and CHR\$ that, in Basic, give my system a shot. You have nothing to lose but your key strokes. Just remember to keep all your escape and control characters between the quotation marks, or the Atari will keep giving you error reports every time you try to enter a line.

MENU

```

420 IF X$(10,10)=" " THEN X$=X$(1,8)
430 IF X$(1,3)="DOS" THEN DOS
440 A$="":FOR I=1 TO LEN(X$):IF X$(I,I)=" " THEN 460
450 A$(LEN(A$)+1)=X$(I,I)
460 NEXT I:X$=A$
470 A$="D1:":A$(LEN(A$)+1)=X$
480 L=LEN(A$):CENTER=9-INT(L/2)
490 GRAPHICS 17:POKE 82,2:POKE 708,26:POKE 710,146
:POKE 712,146:POSITION 5,4:? #6;"LOADING:":POSITIO
N CENTER,6:? #6;A$
500 TRAP 510:RUN A$
510 RESTORE 560:FOR A=START TO FIN:READ VAL:POKE A
,VAL:NEXT A
520 BINFILE=USR(1536,ADR(A$))
530 GRAPHICS 18:POKE 710,146:POKE 712,146:POKE 708
,26
540 POSITION 4,2:? #6;"CANNOT RUN":POSITION CENTER
,5:? #6;A$
550 POSITION 4,8:? #6;"TRY AGAIN":FOR W=1 TO 400:
NEXT W:RUN
560 DATA 162,16,32,173,6,134,207,104,104,157,69,3,
104,157,68,3,169,4,157,74,3,169,3,157,66
570 DATA 3,32,86,228,16,3,76,166,6,169,203,157,68,
3,169,0,157,69,3,169,2,157,72,3,169
580 DATA 0,157,73,3,169,7,157,66,3,32,86,228,16,6,
192,136,240,92,208,96,169,255,197,203,208
590 DATA 4,197,204,240,210,169,205,157,68,3,169,0,
157,69,3,32,86,228,16,2,48,69,165,207,240
600 DATA 14,165,203,141,224,2,165,204,141,225,2,16
9,0,133,207,165,203,157,68,3,165,204,157,69,3
610 DATA 165,205,56,229,203,157,72,3,165,206,229,2
04,157,73,3,254,72,3,208,3,254,73,3,32,86
620 DATA 228,16,137,192,3,240,133,76,166,6,32,173,
6,108,224,2,152,133,212,169,0,133,213,169,12
630 DATA 157,66,3,32,86,228,96

```

□

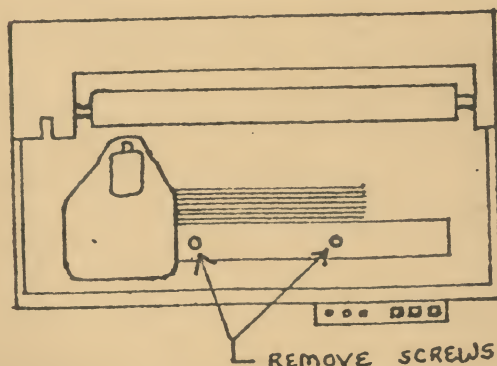


Figure A

NEC Fix

By Jim Wooding

Reprinted from Vol. 3 No. 3

LDMPOC/Santa Maria Atari User Group

Retyped by Danea Koch

I am not the only one to have a high pitch tone (squeal, whine, whistle - call it what you like, emanating from my NEC printer, when the power switch is on and the printer is "idling".

I talked directly with a very helpful NEC technician, the day following the arrival of my 8023, who analyzed my problem. So, with the solution in hand, I made my way to the local electronic parts store and purchased a replacement 200 pf ceramic capacitor. For not telling others, whom I have since found out have been living with the whines and squeals since day one, my sincerest apologies! It seems that a batch of defective capacitors have been installed in many NEC 8023's which do not affect the performance, but do irritate the ears!

The cure is simple enough for anyone who can tell a phillips-head screwdriver from a hammer and who has held hands with a soldering gun. Here is what you will need:

- 1) Phillips screwdriver.
- 2) Needle-nose pliers
- 3) Soldering gun & solder.
- 4) 200 pf ceramic capacitor. (I used a Sprague Q-Line stock number QCP-5135-01 rated at 1000 WVDC +/-20% tolerance.)

The cure is done by:

1. Remove cover (covering the ribbon cartridge & dip switches).
2. Remove two (2) screws as shown in Figure A.
3. Turn printer over, bottom-side up with platen knob facing you.
4. Remove snap-out bottom panel (covering large printed circuit board).
5. Remove only the screws (total 4) located at the corners of the printed circuit board.

CAUTION: DO NOT REMOVE THE REMAINING SCREWS!

6. With the recommended assistance (I didn't have it, but it does make it easier), pull out end nearest you (and farthest from the stepping motor) enough to identify capacitor "C2" (about 1 inch). (See Figure B) This capacitor is marked on the board so you should find it easily. Holding printer bottom-up, with the stepping motor away from you, C2 will be at the extreme right of the printed circuit board, directly behind a similar capacitor. Heat up the leads with your soldering gun, remove the defective capacitor, install the newly-purchased capacitor and re-solder the connections. Assemble in reverse order, plug in and turn on...enjoy the peace and quiet of your NEW NEC!

SWIMMING IN THE DEEP BLUE C
By Howard C. Johnson - JACG

The Deep Blue C is an implementation of the C programming language. It consists of a compiler, linker, and interpreter and is offered by the ATARI Program Exchange as APX-20166. This is a good, usable version of C.

Program preparation is tedious because it is a compiled language. I am spoiled by the immediacy of BASIC; edit, syntax checking, and run, all in one load of the program. C requires four separate functions:

1. Load an editor, and edit those program modules requiring changes.
2. Load the compiler, and compile the changed modules.
3. Load the linker, and link all of the modules comprising the program.
4. Load the program, and run it.

These processes can be exasperatingly slow. The compiler runs slow, but that is to be expected. The real aggravation comes from loading items 2, 3 and 4. These are in normal Save format but have an address section after every 128 bytes which prevents fast loading. Clearly one's second C program should be to copy and compact the compiler and linker load programs.

The ATARI literature recommends a full screen editor for C source code. This is correct, however, I have been able to use the ATARI AS/ED Cartridge as follows: I always edit a line numbered version of the program, PROG.ED. After saving the edited copy (LIST #D:PROG.ED) I also PRINT it (PRINT #D:PROG.C). This strips the line numbers but leaves a space before each line which doesn't bother the Compiler.

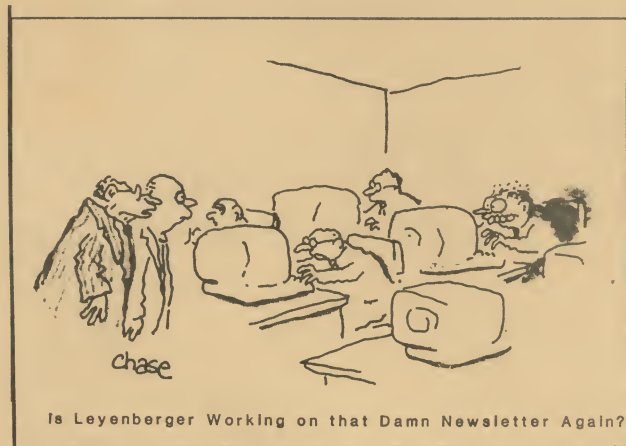
The Linker uses a control file, NAME.LNK, which lists all of the individual modules. The spaces that normally occur after the line numbers must be removed before printing NAME.LNK.

DO's and DON'T's.

1. Keep very few procedures in a file. Compiling is slow, don't waste time recompiling working code.
2. A char data type is always unsigned with values of 0-255. It can never be negative.
3. The status returned after I/O is an integer, which, when the status would have been ≥ 128 is returned as negative. This means that EOF (\$88) is equal to -136.

4. Arithmetic in constant expressions is calculated at run time, instead of being evaluated by the compiler. I.e., `for(i=0;i<5/2+3*7-1;i++)` will run slower than `for(i=0;i<22;i++)`.

Recommendations: This is a good implementation of C. It produces fast running code with only the overhead of 4000 bytes for the interpreter. It is excellent for anyone who wants to learn C at home. Finally, it is great for applications that surpass BASIC but without the mess of writing in Assembler. ▲



J A C G Newsletter Advertisements

The Jersey Atari Computer Group Newsletter will accept advertising on a first-come and space-available basis. Camera ready copy, accompanied by payment, must reach the Editor by the 20th day of the month preceding publication. JACG reserves the right to make decisions concerning the placement of ads within the Newsletter.

Advertising Rates

- | | |
|--------------------------|-------------|
| a) Full page (7-1/2 x 9) | ... \$40.00 |
| b) Half page | ... \$25.00 |
| c) Quarter page | ... \$15.00 |

Discount rates available upon request.

Contact: Herb Lehner
J A C G
1375 Blair Ct.
Bridgewater, N.J. 08807

(201) 582-2774 (days)
(201) 725-9394 (eve.)

**MACE PRESENTS
THE WORLD'S FIRST
ATARI-ONLY COMPUTER CONVENTION**

TARICON '83

**OCTOBER 21-23, 1983
SOUTHFIELD CIVIC CENTER PAVILLION
SOUTHFIELD, MICHIGAN**

***The First Major Consumer Computer Convention
Devoted Exclusively To Atari Home Computers
and Related Support Products***

CONVENTION FEATURES

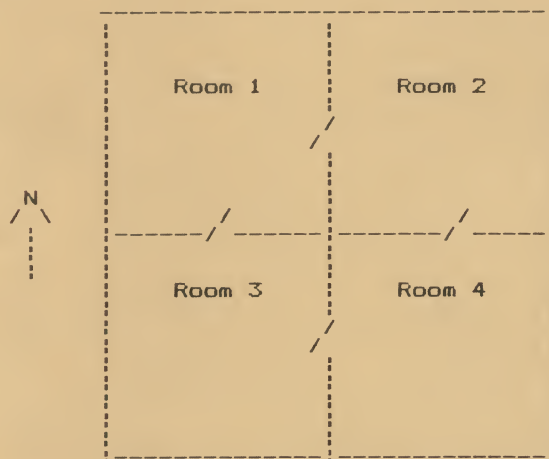
- Over 50 Exhibit Booths with Atari-Related Products
- Giant 20' x 40' Atari Display
- Over 50 Seminars and Workshops Devoted to Helping You Get Maximum Use from Your Atari
- Many, Many Noted Software Authors Speaking on Game, Business and Educational Programs
- Special How-To-Get-Started Seminars
- New Product Introductions
- Game Tournaments
- Seminars/Meetings by MACE's Special Interest Groups
- Pre-Convention Banquet Friday Evening
- Displays by Software, Peripheral and Accessories Companies
- Product Demonstrations and Workshops
- Door Prizes and Raffles
- Thousands of Atari Computer Enthusiasts

For more information, send for our full pre-registration flyer:
TariCon '83 Info, MACE, Box 2785, Southfield, Michigan 48037

Atari is a registered trademark of Atari, Inc.
MACE, Michigan Atari Computer Enthusiasts, an independent non-profit organization devoted to users of Atari home computers,
is not affiliated with Atari, Inc.

Continued from 8

This program assumes a simple room layout such as this:



The first lines set up some constants and direct the player to the first room. After that is the COMMUNICATE routine. The Use routines should always be placed at the front of the program for best response as the system looks through the entire program for the label starting with the first line of the program.

The COMMUNICATE routine takes in the command entered by the operator and then proceeds to break it down to its component parts. We select each word off starting from the left with the MS: (match string) command in line 100. This line looks for the blanks between the words and the result is three strings \$LEFT, \$MID and \$RIGHT which are produced by this operation. They contain as indicated the data to the left, right and match of the search string. As each word is found it is looked up in a list of verbs or actions and if not found there in a list of nouns. If the word is not found in either list then it is ignored. We continue to parse the command entered until there is no match as indicated by the %M variable, a system variable indicating the failure of the match operation by being zero.

With whatever has been found for the verb and noun variable we create a command in the format that we can check and then return to the place that this routine was called.

Once in the room routine you can do many things. Most obviously expand the room description. This can be a full screen or more if necessary. Multiple screens can be controlled by asking the player to respond to an Accept command to cause the next page to appear. This area is where the creative writing comes in, have each student prepare a room description for the game and then enter it into the game. Have each student enter the description to the system and save it as a file with predetermined line numbers so it will not conflict with the other modules. Then when all the rooms are done

merge them together by loading them all into memory and then save the game as one member. Other variations for this area are to have the descriptions given in foreign languages, have people describe a historic place or country for the other student to guess, have students create a "map" of the town or school just by description. You are only limited by the memory of your system. Once in the room module match the command to the command list for the room. The match list is paired to a list of Jumps for each match. Include routines for the moves not allowed and for the command not understood so that the player is given some feedback before redisplaying the description. As you allow more verbs and nouns then these match lists must be expanded. If one line will not hold all the data necessary then create more match and jump lists.

To expand on the mechanics of grammar and provide an exercise in communication skills expand the interpreter or COMMUNICATE routine of this program to allow a wider variety of words and to allow abbreviations such as N, S, E or W for the compass headings. Experiment with the program I've given you here. This is not the only or best solution to this problem. I have also left off the part of the program for the other two rooms in the diagram, try expanding the program by adding the other two rooms following the format of the first two.

THE G.I.T.-1 POWER PACK Reviewed by Dick Kushner-JACG

The G.I.T.-1 (for Get-It-Together) Power Pack is a box containing switched outlets for four Atari power supplies, three AC outlets, varistor surge protection and a two position on/off switch. The advantage of this little goodie is that it eliminates the tangled mess of all those Atari power supply cords— one for the computer, one for each disk drive, one for the 850 interface, etc. In one compact box, G.I.T.-1 provides power supplies to run all these peripherals and also three AC outlets to plug in the TV and printer and modem. Now only one power cord runs to the wall outlet. The prototype that I have been using for a month does not have lighted switches to show which power supplies are active, but I am told that the final product will. The main power switch can be set to turn on only the power supplies or to turn these and the AC outlets on, a nice little touch. For more information, contact JACG member Tony Pellechio (201-534-4404).



Tech Corner
 32767 POKE
 (PEEK(138)+256*PEEK(139)+2),0:SAVE
 "D:UNLSTABL":NEW

LISTING #2:SOURCE TO UNPROTECT A SAVED PROGRAM
 ENTER THIS PROGRAM AND RUN IT SPECIFYING THE NAME OF THE PROGRAM TO BE PROTECTED.
 NOTE: HAVE A BACK-UP COPY AVAILABLE; JUST IN CASE

```

10000 OPEN #6,12,0,"E:":POKE 710,0:DIM
FN$(15)
10100 ?CHR$(125);"FILENAME:":INPUT
#6;FN$:OPEN #1,12,0,FN$
10200 GET #1,X1:GET #1,X2:CNTR=3
10300 IF X1=255 THEN 10500
10400 X1=X2:GET #1,X2:CNTR=CNTR+1:GOTO
10300
10500 IF X2=127 THEN 10700
10600 GOTO 10400
10700 WMARK=CNTR
10800 IF X1=0 THEN 11000
10900 X1=X2:GET #1,X2:CNTR=CNTR+1:GOTO
10800
11000 IF X2=128 THEN 11200
11100 GOTO 10900
11200 WVAL=CNTR-WMARK
11300 ? "ZAP LOCATION ";WMARK;" TO ";WVAL
11400 CLOSE #1:OPEN #1,12,0,FN$
11500 FOR A=1 TO WMARK-1:GET #1,X:NEXT A
11600 PUT #1,WVAL:CLOSE #1
  
```

Next month we will take a look at ATARI's disk operating system. I plan to spend several issues on this topic so that even novice programmers will be able to understand how the disk does what it does. The discussion of DOS will pertain to release 2.05. Several programs will be presented to aide in diagnosing disk problems, and how to perform 'impossible' tasks with the 810

Until next month remember- Always PEEK before you POKE

Languages

use. String variables must be declared and DIMensioned before. Microsoft does not require DIMensions unless there are more than 10 places required.

Operational Symbols: BASIC uses the normal arithmetic symbols + - * / ^ and relational operators < > = <> <= >= . Many BASICS also use the Boolean operators AND, OR, NOT.

Program control: Program control statements are used to direct the flow of operations within a program. Examples would be FOR...NEXT, IF...THEN, ON...GOTO .

Subroutines: A subroutine is a relatively independant portion of the program. It need only be written in the program once and can be used as often as necessary. The program departs for a subroutine and returns when the subroutine is completed. The most common examples are GOSUB/RETURN and GOTO.

Direct Memory Access: Personal computers allow the use of PEEK and POKE statements for direct access to memory locations. Memory locations in most systems using BASIC are one byte in size, meaning they can hold a number from 0-255. A number from 0-255 can be POKE'd into a memory location, or we can PEEK into the contents of a memory location.

Higher level languages such as BASIC have some disadvantages when compared to lower level (machine) languages that are used by the microprocessor. Machine language allows certain operations to be done faster than in BASIC and it also allows the performance of some functions that are difficult to do in BASIC.

BASIC allows subroutines that are written in machine language to be included within the body of the program. These subroutines can be accessed by a USR statement. Program control will return to BASIC when a RETURN command is reached in the machine language subroutine.

Program Editing: BASIC includes several features that are designed to simplify editing and changes. Some of these are AUTO,CLEAR, CONT, DELETE, RENUM.

In general I find BASIC to be a good, compact language suitable for the home user. When applied to business or scientific applications it is weak in file handling. It is also a general purpose language not suited to every application.

If you or your children will be using BASIC on other computers such as the APPLE, TRS, etc., I strongly recommend ATARI Microsoft BASIC. Obviously written by Microsoft Corp, it adds many important features that you will see on other machines for commercial applications, while maintaining the capabilities for ATARI sound and graphics. It also contains appendices for converting TRS-80,

Applesoft and ATARI BASIC programs to Microsoft.

Since I write many financial programs, my favorite feature is the PRINT USING command, which allows for line and column formatting just like the COBOL commands. It is important to note that Microsoft is not interchangeable with ATARI BASIC, you must choose one or the other. A major difference comes with PRINT statements. Microsoft does not use LPRINT, you must open a channel, and PRINT to it. There are advantages and disadvantages to this method, but the user must be aware of the structural differences.

That's it for this month. Next time I will take a look at programming in Pascal, comparing it to the BASIC language.

Continued from

The last bits of information I would like to share with you have to do with using the printer and not Letter Perfect. The Leading Edge manual has some typos in it. Left margin set is CHR\$(27), CHR\$(76), backspace only works in Incremental Print Mode and the fact that the program for Electronic Vertical Format Unit (EVFU) does not work, even for the TRS-80 computer it was configured for. Again I contacted Leading Edge and they admitted this program does not work and supplied me with a simple program that is not an EVFU but just a program to skip lines and does not program the printer to tab vertically and horizontally automatically when using forms. If the solution can be found that allows us to use this feature please pass it on. △

HAPPY 810 ENHANCEMENT
Reviewed by W. Chin - JACG
Copyright (c) 1983 W. Chin

The disc drive is a considerably faster way of storing and accessing data than the cassette recorder. Now there is a device that claims to make the drive load and save at an even faster rate. This add-on device is manufactured by Happy Computing which is located in California. The north Jersey distributor is Gemini Enterprises in Cedar Knolls.

For the \$200+ dollars you spend, you get a circuit board with several chips soldered directly on it plus a diskette which will enable you to copy any diskette including specially formatted ones, control the reading rate of the drive and test the board. In terms of electronics, what you are doing is replacing the ROM chip with an improved ROM program. This eliminates the need for a fast format chip. A data separator circuit is included in the board. This means that if you do not have one of the newer drives with a data separator, you will be updating it.

You need not be an expert to install this board since no soldering is necessary. (With the earliest model there was some required.) The most difficult aspect of the assembly procedure is to line up the 28 and 40 pins located on the board to the mother board so no pins get bent when pushing the 68 pins into the two sockets. Other than the removal of three ICs the job is simple.

I timed the amount of running time the modified drive took to do several different types of programs. For each set of tests I tested the speed of the drive before and after to make sure the drive was still running at 288 RPMs. I obtained the following results:

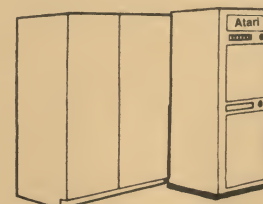
HAPPY VS. 810

| | | |
|-----------------|--------|--------|
| Atari Dos II | 15sec. | 16sec. |
| Letter Perfect | 11sec. | 12sec. |
| Atari Word Pro. | 24sec. | 28sec. |
| Pharaoh's Curse | 32sec. | 44sec. |

This add-on device is worth the investment if you do not like waiting for information to be saved or loaded. The drive's actual running time is less with this board. The typical snorking that occurs when reading bad sectors is diminished. Even if a program must be run with a 'forced' slow mode speed there can be a decrease in the number of mild snorkings that occur. An example is Pharaoh's Curse which when loaded normally snorks four times. With the forced slow mode reading the snorking is dramatically milder and there are only three of them. Another example is APX's Fixdisk. When dumping out sectors the drive is continuously running. With the Happy the drive has time to stop. The reason for this is the existence of a buffer in the modification and the capability of reading several sectors quickly.

The increase in speed does not come from the motor running faster. It comes from the Happy reading tracks instead of reading one sector at a time. In terms of its ability to copy any disk, I have yet to be disappointed. One should keep in mind that some programs will not load properly with the higher rate of loading. If you do make a copy of a program that will not run at the higher speed, it is possible to program that copy to tell the drive to read at normal speed and the drive will do so. Overall, I am pleased with the Happy though the speed increase is useful (1) in loading games with bad sectoring or unusual formats (2) for loading long programs. The biggest advantage probably is the prolonging of the existence of my drive by decreasing the amount of time it must work.

[Ed. Note: JACG members are reminded that making copies of copyrighted software for purposes other than personal backup is strictly illegal.] △



COMPUTER
PROCESSOR

Fed up with Frogger? Sick of Star Raiders? Bored with Basic? Addled by Assembler? Well, I've just found the cure for your ills. After a long days work, a couple games of Defender, and a little programming, wouldn't it be nice to sit down and go to another Universe? I know, that's a tough question, but don't worry, Starcross, from Infocom, supplies all you need to get there.

Yes, Starcross is the game that comes in that Frisbee type package. Infocom has probably chosen this sort of packaging to get your attention, as it probably did. Now, with the "Frisbee" type package, includes many other pieces of information. Also included are a Galactic Space Chart and an instruction booklet very well written (in Infocom tradition) which explains all necessary operations in the Science-Fiction type adventure game.

You begin your mission in the year 2186, lying in your bunk on the spacecraft Starcross. You hear an alarm ring, and proceed to investigate its source. You find that you're drifting in space, looking for Black Holes, your planet's source of energy. Quickly you look at your computer's CRT. Yes, it is a Black Hole! With no delay, you begin to read off coordinates to your ships computer and you arrive at your destination seconds later.

Hold on a second, this isn't a Black Hole, it's a Space Habitat, a tiny world floating in Space. Once landed and tied down, you leave your ship, only to be confronted with a most puzzling puzzle. This puzzle is only the first of many, which wait you inside. Once inside you must find different color rods, and place them in the appropriate colored slots. You say this is easy? Wrong! The game is filled with puzzles, creatures, robots, different rods, and much, much more. No words can describe the complexity of this "little" world.

Starcross, as all other Infocom programs, is a piece of programming genius. One can enter full English sentences, and communicate easily with all other inhabitants of the little world. Technically, the game is A+, with a save game feature and many others which allow error free operation. The program requires a disk drive and 32K.

If you're fed up with Frogger, sick of Star Raiders, bored of Basic, or addled by Assembler, buy Starcross, a truly fine adventure program, which as the ad says, puts graphics where the sun don't shine - your imagination.

In conjunction with our July meeting on communications services, I thought I would provide some additional information for modem T/C users. Data bases were not initially formed for home users, but home computer use is certainly foremost in the minds of many of these companies. The reason is the size of the potential market, and the fact the the P/C user would take advantage of non-prime time. The rates the DB companies charge after 5 PM are much lower than 9-5, but it is almost all profit.

A brief summary of some DataBases:

Dow Jones: offers business and economic news, stock quotes, investment information. Started in 1977; has 60,000 subscribers

Compuserve: offers financial information, banking, games, shopping, weather, encyclopedia, BBS, and more. Started 1979; 36,000 subscribers. It was Compuserve who gave a demo at our Jan 83 meeting.

The Source: News and research tools, consumer tips, market quotes, travel, shopping, BBS. 26,000 subscribers.

NewsNet: More than 80 newsletter covering many subjects. Began Apr 1982

Dialog Information Services: Descriptive indexes of articles, books, and newspapers. Begun 12/82

DRI-VisiCorp: 58 Updated Databases; economic forecasts, foreign trade, industrial. Begun 1/83

There are more than space permits me to list, and I have purposely omitted prices because I only subscribe to Compuserve which is \$5 per hour non-prime. Most have a one time connection charge ranging from \$20 to \$100. There is solid backing behind the the largest. Dows Jones speaks for itself; H&R Block owns Compuserve and Reader's Digest owns The Source. DRI (Data Resources Inc) is a joint effort of McGraw-Hill and Visi-Corp, offering Visi-Link economic and stock information.

Bulletin Boards (BBS): If free is within your price range, then try one of the many BBS available such as the JACG BBS. Many of the larger clubs have data bases offering free games and utilities, as well as a message 'bulletin board' system for members to communicate. The MAY JACG newsletter list some of the ATARI club BBS.

□

COIN PROGRAM
by Lora & Steve Van Slyke

Program Summary

Have you ever wanted to keep track of your coin collection by the date? This program was designed to do just that. It's menu supplies the user with options of Input, Display, Save, Retrieve, and End the program. Starting with the input; it will ask you for the date of your coin; however, it is limited to the dates of 1900 to 1999 for inputs. If you have finished inputting the dates just press <RETURN> twice after last date and you will proceed back to the menu. You can then save your data on a disk, or retrieve data if you wish to expand your collection. In the display mode, you will get a number of how many coins you have on a certain date. Columns by decades, rows by single years and then at the bottom of the screen you will get the total accumulated coins.

Programming Techniques

The count of coins is in a single subscript array called P (line 70). Also the menu response (lines 162 and 350) is checked for other inputs besides those used. The display takes a new approach (lines 990-1210) because it prints out vertically instead of horizontally. It will also add and display the total coins in collection (lines 1230-1250). There is a menu (lines 156-162) that gets type of coins being used. A record check is built in, just in case there are no records to save (lines 680,685). There is another check in case the record is full (line 585).

Program Adaptations

If you have a cassette, instead of a disk, change line 70 to DIM P(99), F\$(2), CAT\$(8): F\$="C:" and delete line 163.

```

10 REM %COINS PROGRAM%
30 REM %BY LORA & STEVE VAN SLYKE%
50 REM % 4-2-83 VERSION 1.6 %
70 DIM P(99),F$(7),CAT$(8),X$(1):F$="D:--DAT"
90 FOR D=0 TO 99:P(D)=0:NEXT D
110 REM %%%%%%%%%%
130 REM %MENU%
150 REM %%%%%%%%%%
155 GRAPHICS 0
156 ? "Enter the coin category:"
157 ? :? :? :? :? :? " 1.....Pennies"
158 ? " 2.....Nickels"
159 ? " 3.....Dimes"
160 ? " 4.....Quarters"
161 ? " 5.....Half Dollars"
162 L=155:TRAP 4000:INPUT C1:IF C1<1 OR C1>5 THE
N GOTO 155
163 F$(3,3)=CHR$(64+C1)
169 FOR N=1 TO C1:READ CAT$:NEXT N
170 GRAPHICS 0
190 ? "Coin Collection Program"
210 ? :? :? :? "Your options are:"
230 ? " 1.....Display"
250 ? " 2.....Input"
270 ? " 3.....Save"
290 ? " 4.....Retrieve"
310 ? " 5.....End program"
330 ? :? :? :? "What is your choice";
350 L=170:TRAP 4000:INPUT C:IF C<1 OR C>5 THEN 1
70

```

```

370 ON C GOTO 930,480,680,830
410 END
430 REM %%%%%%%%%%
450 REM %INPUT%
470 REM %%%%%%%%%%
480 GRAPHICS 0
490 TRAP 610: ? :? CAT$;": What is the date , 19"
:INPUT D
570 IF D<0 OR D>99 THEN ? "Invalid input":GOTO 4
90
585 IF P(D)>254 THEN ? "The record is filled for
that date.":GOTO 490
590 P(D)=P(D)+1:GOTO 490
610 L=530:TRAP 4000:GOTO 170
630 REM %%%%%%%%%%
650 REM %SAVE DATA%
670 REM %%%%%%%%%%
680 FOR X=0 TO 99:IF P(X)>0 THEN 690
685 NEXT X: ? "THERE ARE NO RECORDS":FOR G=1 TO 4
00:NEXT G:GOTO 170
690 OPEN #1,0,0,F$
710 FOR X=0 TO 99
730 A=P(X):PUT #1,A
750 NEXT X:CLOSE #1:GOTO 170
770 REM %%%%%%%%%%
790 REM %RETRIEVE DATA%
810 REM %%%%%%%%%%
830 TRAP 910
850 OPEN #1,4,0,F$
870 FOR X=0 TO 99
890 GET #1,A:P(X)=A:NEXT X
910 CLOSE #1:TRAP 4000:GOTO 170
930 REM %%%%%%%%%%
950 REM %DISPLAY%
970 REM %%%%%%%%%%
990 GRAPHICS 0:TD=0
1005 X=INT(20-(LEN(CAT$)/2)):FOR Y=1 TO X: ? " ";
:NEXT Y: ? CAT$
1010 TC=-20
1030 FOR C=3 TO 30 STEP 3:POKE 82,C:POKE 84,2:TC
=TC+10: ?
1070 IF SGN(TC)=-1 THEN ? " ":GOTO 1130
1090 IF SGN(TC)=0 THEN ? TC;TC:GOTO 1130
1110 ? TC
1130 FOR U=0 TO 9
1150 IF SGN(TC)=-1 THEN ? U:GOTO 1190
1170 ? P(TC+U)
1190 NEXT U:POKE 84,0
1210 NEXT C
1230 FOR X=0 TO 99:TD=TD+P(X):NEXT X
1250 POKE 82,0:POSITION 0,15: ? " TOTAL COINS:";T
D: ? :? :? " PRESS <RETURN> TO GO BACK TO THE MEN

```

```

U":INPUT X$:GOTO 170
2000 DATA PENNIES,NICKELS,DIMES,QUARTERS,HALFS
4000 ? "INVALID INPUT TRY AGAIN ":FOR D=1 TO 400
:NEXT D:TRAP 4000:GOTO L

```


ZORK II
Reviewed by Eric Brown - JACG

After slaving over a hot computer to finish Zork I, why would anyone in their right mind want to buy Zork II? Well, I was in my right mind, but I bought it anyway! Maybe spending close to six months on Zork I wasn't enough!

"The next step downward to danger."

Would you shell out thirty bucks for a program that had that on its cover? Well, if you're adventuresome and you had the thrity bucks, sure you would. That's right, I'm reviewing another Infocom game, Zork II.

Booting up with Zork II is extremely easy. Just turn on the drive, pop in the disk, remove any cartridges, and flip the switch on your computer. As always, Infocom has spared no expense in their distributing of this adventure. In other words, the disk is reliable. I can vouch for this because I've almost completed Zork II after about six months work. First off, let me warn you that wandering in Zorkland is not a picnic for your disk drive. Every move, no matter what, the disk drive spins and beats on that disk. I personally have had no problems, but I wouldn't want to be blamed for a messed up drive head.

Now some more bad news. Zork II is not easy! It, like all other Infocom programs, is all text. Your "mission" is to discover treasures and bring them back to a specified place. The treasures are guarded by robots, cages, wizards, and the like. I don't want to scare you off, but this is truly an advanced adventure. It won't be solved in a day or two. Let's put it this way. I bought it in August of 1982 and am almost no where near finishing it today. I, however, haven't bought a map or hintbook. Yes, maps will be available for a short time. The Zork Users' Group (ZUG) that offers these detailed maps and clues is going out of business very soon. Infocom, however, is going to continue to supply maps later in the year.

On all technical aspects this adventure deserves its high acclaim. The program responds to almost any reasonable English sentence making it very easy to communicate with Zork. One example of a reasonable sentence Zork will accept is "TELL ROBOT 'PULL SWITCH'." As you can see, this is obviously higher quality than Scott Adams' adventures. However, Zork unlike The Adams series, requires 32k memory or more. Other features include printing descriptions to a printer and saving and restoring up to five saved game positions on one disk. For saved games one must use a blank disk of his own.

Overall I was very satisfied with Zork II. It held my attention for quite some time and is "different" from those fly-by night arcade games.

ATARI Microsoft BASIC II
Reviewed by Arthur Leyenberger - JACG

The cartridge version of Microsoft BASIC (AMSB) has just been released by Atari. For \$79.95 (list) you get a 16K cartridge, an extension diskette, a spiral-bound user manual, quick reference guide and brief overview.

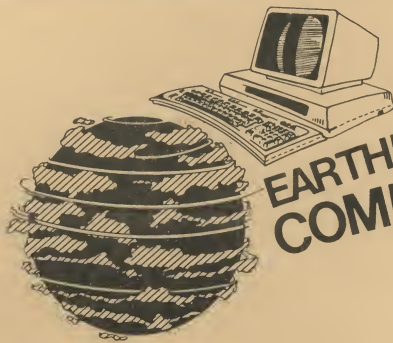
First of all, when the cartridge and the disk are used together, this version of Microsoft BASIC is exactly the same as the previous disk-based version which was released about two years ago. The main difference is that Atari was unable to fit 18K bytes worth of AMSB into a 16K byte cartridge (understandably enough). Therefore, the disk contains another 2K bytes worth of "extension" features. Fortunately, the extension disk is copyable under DOS so a backup can be made.

There are ten commands that did not make it into the cartridge. They are: AUTO - for automatic line numbering; DEL - for deleting individual or blocks of lines; RENUM - for renumbering lines; TROFF and TRON - for tracing (on and off) the sequential flow of a program; NAME...TO - for re-naming disk files from BASIC itself; VERIFY - for comparing a disk file with the current contents of memory; DEF - for defining numeric or string functions; NOTE - for obtaining the disk sector number and byte count of a DOS file; and PRINT USING - for formatted screen or printer output. Seven of these commands are development aids while three (DEF, NOTE and PRINT USING) are used from within a program. However, once the extension disk is booted up, the entire language is memory resident.

The other major difference between this and the previous version of AMSB is the documentation. The user manual accompanying the previous release was adequate. The new documentation is excellent. Included within the user manual are tutorial sections on topics such as player/missile graphics and character graphics. The new user manual is also easier to understand. The PRINT USING statement was not covered very well in the old manual and was somewhat confusing. The section in the new manual pertaining to this command is much clearer and includes more examples. Especially helpful is the separate quick reference guide which is well organized and clearly presented.

In conclusion, the Atari Microsoft BASIC II language, documentation and packaging are well done. For the serious ATARI computer programmer, AMSB is the way to go.

I would like to thank Gemini Enterprises for loaning me a copy of this software for review.



EARTHRISE COMPUTER CENTER

Epson RX 80 \$ 369



100cps QUAD DENSITY GRAPHIC MODE / TRACTOR FEED ONLY
ELITE CHARACTER SET.

DISKS

ELEPHANT DISKS
3M Scotch S/S,D/D

\$ 20.95 per box
\$ 27.95 per box



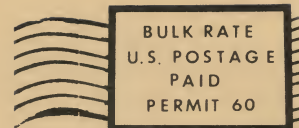
ALL PRICES CASH AND CARRY ADD 3% MC,VISA Prices effective through
8/22/83

ATARI 400, BASIC & RECORDER

\$ 199

after \$50.00 Rebate from Atari

 * J A C G *
 * JERSEY ATARI COMPUTER GROUP *
 * 58 DEWEY AVENUE *
 * HIGH BRIDGE. NEW JERSEY 08829 *



JACG NEWSLETTER - VOLUME 2, Number 12
 August 1983

J A C G OFFICERS

President: Richard Kushner
 58 Dewey Ave.
 High Bridge, N. J. 08829
 (201) 638-8732 (home)
 (201) 582-4794 (work)

Librarian: Don Ursem (Disk)
 37 Clover Lane
 Randolph, N.J. 07869
 (201) 895-2522 (home)

Treasurer: Rick Olson
 5 Starling Drive
 Randolph, N.J. 07869
 (201) 366-6862 (home)

Vice Pres: Dennis Kushler
 45 Mountain Ave.
 Summit, N. J. 07901
 (201) 273-7588 (home)

Secretary: Dave Logothetis
 (201) 372-5272 (home)

Programs: Richard Rospond
 (201) 635-2936

Editor: Arthur Leyenberger
 40 Lawrence Rd.
 Parsippany, N. J. 07054
 (201) 887-2861 (home)

Advertising: Herb Lehner
 1375 Blair Ct.
 Bridgewater, N. J. 08807
 (201) 725-9394 (home)

The Jersey Atari Computer Group (JACG) is an independent, informal organization of ATARI computer users. It is not affiliated with Atari or any other commercial enterprise. Opinions expressed in this publication reflect only the views of the individual author, and do not necessarily represent the views of JACG. Material in this Newsletter may be reprinted by other Atari User Groups, provided the author (if applicable) and JACG are given credit and the issue date is given. Only original work may be reprinted. Questions concerning reprinting should be addressed to the Editor.